NATIONAL SENIOR CERTIFICATE EXAMINATION
NOVEMBER 2017

INFORMATION TECHNOLOGY: PAPER II

MARKING GUIDELINES

Time: 3 hours 120 marks

These marking guidelines are prepared for use by examiners and sub-examiners, all of whom are required to attend a standardisation meeting to ensure that the guidelines are consistently interpreted and applied in the marking of candidates' scripts.

The IEB will not enter into any discussions or correspondence about any marking guidelines. It is acknowledged that there may be different views about some matters of emphasis or detail in the guidelines. It is also recognised that, without the benefit of attendance at a standardisation meeting, there may be different interpretations of the application of the marking guidelines.

SECTION A

1.1
```
SELECT *
FROM tblProfiles
ORDER BY DailyPostRate DESC;
```

1.2
```
SELECT Username, Fullname
FROM tblProfiles
WHERE Privacy="public";
```

1.3
```
INSERT INTO tblMembership (GroupID, ProfileID, Admin, Posts)
VALUES (8, 98, TRUE, 0);
```
(Fields do not need to be listed)

1.4
```
SELECT GroupID, ROUND(AVG(Posts) , 1)
FROM tblMembership
GROUP BY GroupID;
```

1.5
```
SELECT MONTH(CreatedDate) AS CreatedMonth, COUNT(*) AS
TotalGroups
FROM tblGroups
GROUP BY MONTH(CreatedDate)
HAVING COUNT(*) > 2;
```
Allocate this mark for correctly rename fields CreatedMonth and TotalGroups

Allocated this mark for correctly using MONTH and COUNT in the SELECT clause

1.6
```
SELECT GroupID; SUM(DailyPostRate * 7) AS PredictedWeeklyPosts
FROM tblProfiles, tblMembership
WHERE tblProfiles.ProfileID = tblMembership.ProfileID
GROUP BY GroupID;
```

1.7
```
SELECT GroupName, COUNT(*)
FROM tblProfiles, tblGroups, tblMembership
WHERE tblProfiles.ProfileID = tblMembership.ProfileID AND
tblGroups.GroupID = tblMembership.GroupID AND Admin = true AND
Privacy = "private"
GROUP BY GroupName;
```

1.8
```
UPDATE tblGroups SET Icon = LEFT (Icon, LEN(Icon) – 5)&".jpg"
WHERE RIGHT(Icon, 4) = "jpeg";
```


SECTION B OBJECT-ORIENTED PROGRAMMING

JAVA SOLUTION:

QUESTION 2

```
// Question 2.1
public class Post
{
  // Question 2.2
  private int id;
  private String user;
  private String postContent;
  private String postDateTime;

  // Question 2.3
  public Post (int inID, String inPostUser, String inPostContent, String
                        inPostDateTime)
  {
    id = inID;
    user = inPostUser;
    postContent = inPostContent;
    postDateTime = inPostDateTime;
  }

  // Question 2.4
  public int getPostID()
  {
    return id;
  }

  // Question 2.5
  public String toString()
  {
    return postDateTime + " " + user + " posted: " + postContent;
  }

}
```

Private
Correct Types
Appropriate names

Assign correct values to attributes

Named correctly, return type is correct, and returns the correct attribute value

QUESTION 3

```
// Question 3.1
public class Response
{
  // Question 3.2
  private int postID;
  private String user;
  private String comment;
  private String responseDateTime;
  private int responseType;

  // Question 3.3
  public static final int RESPONSE_LIKE = 1;
  public static final int RESPONSE_COMMENT = 2;
  public static final int RESPONSE_DISLIKE = 3;
  public static final int RESPONSE_LOVE = 4;

  // Question 3.4
  public Response(int inPostID, String inUser, String inComment, String
                          inResponseDateTime, int inResponseType)
  {
    postID = inPostID;
    user = inUser;
    comment = inComment;
    responseDateTime = inResponseDateTime;
    responseType = inResponseType;
  }

  // Question 3.5
  private String getResponseType()
  {
    switch (responseType)
    {
      case RESPONSE_LIKE:
        return "likes";
      case RESPONSE_COMMENT:
        return "commented";
      case RESPONSE_DISLIKE:
        return "dislikes";
      case RESPONSE_LOVE:
        return "loved";
      default:
        return "";
    }
  }

  // Question 3.6
  public int getPostID()
  {
    return postID;
  }

  // Question 3.7
  public String toString()
  {
    String rString = responseDateTime + "\t" + user + " " + getResponseType();

    if (responseType == RESPONSE_COMMENT)
    {
      return rString + " on this post: " + comment;
    }
    else
    {
      return rString + " this post";
    }
  }
}
```

Private
Correct Types
Appropriate names

Static/Constants
Named correctly
Integer values assigned correctly

Response Type parameter is an integer

Correct string returned based on integer value

Named correctly, return type is correct, and returns the correct attribute value

Comments dealt with correctly

QUESTIONS 4 AND 7.1

```
import java.io.FileReader;
import java.util.Scanner;

// Question 4.1
public class InstaPageManager
{
  // Question 4.2
  // 5 marks
  private Post posts[] = new Post[100];
  private Response responses[] = new Response[500];
  private int postCounter = 0;
  private int responseCounter = 0;

  // Question 4.3
  public InstaPageManager()
  {
    try
    {
      Scanner fin = new Scanner(new FileReader("data.txt"));

      while (fin.hasNextLine())
      {
        String line = fin.nextLine();

        Scanner tokens = new Scanner(line).useDelimiter("#");

        int id = tokens.nextInt();
        String user = tokens.next();
        String content = tokens.next();
        String dt = tokens.next();

        if (tokens.hasNextInt())
        {
          int responseType = tokens.nextInt();
          responses[responseCounter] = new Response(id, user, content, dt,
                        responseType);
          responseCounter++;
        }
        else
        {
          posts[postCounter] = new Post(id, user, content, dt);
          postCounter++;
        }
      }

      fin.close();
    }
    catch (Exception e)
    {
      System.out.println(e);
    }
  }
```

Counters declared correctly
Private

Split into tokens

Inserted into correct array
Increment Counter

```
  // Question 4.4
  public String getAllPosts()
  {
    String rString ="";
    for(int i = 0; i < postCounter; i++)
    {
      rString += posts[i].toString() + "\n";
    }

    return rString;
  }



  // Question 7.1
  private PostWithResponse prArray[];

  public String combinePostsAndResponses()
  {
    String rString = "";
    prArray = new PostWithResponse[postCounter];
    for(int i = 0; i < postCounter; i++)
    {
      Response tmpResponses[] = new Response[100];
      int tempCounter = 0;
      for (int j = 0; j < responseCounter; j++)
      {
        if (posts[i].getPostID() == responses[j].getPostID())
        {
          tmpResponses[tempCounter++] = responses[j];
        }
      }

      Response postResponses[] = new Response[tempCounter];
      System.arraycopy(tmpResponses, 0, postResponses, 0, tempCounter);
      prArray[i] = new PostWithResponse(posts[i], postResponses);
      rString += prArray[i].toString() + "\n\n";
    }

    return rString;
  }

}
```


QUESTIONS 5 AND 7.2

```
// Question 5.1
public class InstaPageUI
{
  public static void main(String[] args)
  {
    // Question 5.2
    InstaPageManager ipm = new InstaPageManager();
    // Question 5.3
    System.out.println(ipm.getAllPosts());


    // Question 7.2
    System.out.println(ipm.combinePostsAndResponses());

  }

}
```

QUESTION 6

```
// Question 6.1
public class PostWithResponse
{
  // Question 6.2
  private Post post;
  private Response responses[];

  // Question 6.3
  public PostWithResponse(Post inPost, Response[] inResponses)
  {
    post = inPost;
    responses = inResponses;
  }

  // Question 6.4
  public String toString()
  {
    String rString = post.toString() + "\n\t\tReactions: ";

    for(int i = 0; i < responses.length; i++)
    {
      rString += "\n\t\t" + responses[i].toString();
    }

    return rString;
  }
}
```

DELPHI SOLUTION:

QUESTION 2

```
unit uPost;

interface

// Question 2.1
type TPost = class
 // Question 2.2
 private
  id : integer;
  user, postContent, postDateTime : String;
 public
  constructor Create(inID : integer; inPostUser, inPostContent, inPostDateTime
                        : String);
  function getPostID : integer;
  function toString : String;
end;

implementation

{ TPost }

// Question 2.3
constructor TPost.Create (inID: integer; inPostUser, inPostContent,
 inPostDateTime: String) ;
begin
 id := inID;
 user := inPostUser;
 postContent := inPostContent;
 postDateTime := inPostDateTime;
end;

// Question 2.4
function TPost.getPostID: integer;
begin
 Result := id;
end;

// Question 2.5
function TPost.toString: String;
begin
 Result := postDateTime + ' ' + user + ' posted: ' + postContent;
end;

end.
```

Private
Correct Types
Appropriate names

Assign correct values to attributes

Named correctly, return type is correct, and returns the correct attribute value

QUESTION 3

```
unit uResponse;

interface

// Question 3.1
type TResponse = class
 // Question 3.2
 private
  postID : integer;
  user, comment, responseDateTime : String;
  responseType : integer;
 public
  constructor Create(inPostID : integer; inUser, inComment, inResponseDateTime : String;
                          inResponseType : integer);
  function getResponseType : String;
  function getPostID : integer;
  function toString : String;
end;

// Question 3.3
const
  RESPONSE_LIKE = 1;
  RESPONSE_COMMENT = 2;
  RESPONSE_DISLIKE = 3;
  RESPONSE_LOVE = 4;


{ TResponse }
implementation

// Question 3.4
constructor TResponse.Create(inPostID: integer; inUser, inComment,
 inResponseDateTime: String; inResponseType: integer);
begin
 postID := inPostID;
 user := inUser;
 comment := inComment;
 responseDateTime := inResponseDateTime;
 responseType := inResponseType;
end;

// Question 3.5
function TResponse.getResponseType: String;
begin
 case responseType of
  RESPONSE_LIKE: result := 'likes';
  RESPONSE_COMMENT: result := 'commented';
  RESPONSE_DISLIKE: result := 'dislikes';
  RESPONSE_LOVE: result := 'loved';
  else result := '';
 end;
end;

// Question 3.6
function TResponse.getPostID: integer;
begin
 Result := postID;
end;
```

Private
Correct Types
Appropriate names

Static/Constants
Named correctly
Integer values assigned correctly

Response Type parameter is an integer

Correct string returned based on integer value

named correctly, return type is correct, and returns the correct attribute value

```
// Question 3.7
function TResponse.toString: String;
begin
 Result := responseDateTime + #9 + user + ' ' + getResponseType;

 if responseType = RESPONSE_COMMENT then
  begin
   Result := result + ' on this post: ' + comment;
  end
 else
  begin
   Result := result + ' this post';
  end;
end;

end.
```

Comments dealt with correctly

QUESTIONS 4 AND 7.1

```
unit uInstaPageManager;

interface

uses uPost, uResponse, uPostWithResponse, Dialogs, SysUtils;

// Question 4.1
type TInstaPageManager = class
 private
  // Question 4.2
  posts : array[1..100] of TPost;
  responses : array[1..500] of TResponse;
  postCounter, responseCounter : integer;
  // Question 7.1
  prArray : array of TPostWithResponse;
 public
  constructor Create;
  function getAllPosts : String;
  function combinePostsAndResponses : String;
end;

implementation

{ TInstaPageManager }

// Question 4.3
constructor TInstaPageManager.Create;
var
 inFile : TextFile;
 line, user, content, dt : String;
 postID, rType : integer;
begin

 postCounter := 0;
 responseCounter := 0;

 If FileExists('data.txt') then
  begin
   AssignFile(inFile, 'data.txt');
   Reset(inFile);

   while NOT EOF(infile) do
    begin
     ReadLn(inFile, line);
```

```
     postID := StrToInt(Copy(line, 1, Pos('#', line) - 1));
     Delete(line, 1, Pos('#', line));

     user := Copy(line, 1, Pos('#', line)-1);
     Delete(line, 1, Pos('#', line));

     content := Copy(line, 1, Pos('#', line)-1);
     Delete(line, 1, Pos('#', line));

     if Pos('#', line) > 0 then
      begin
       dt := Copy(line, 1, Pos('#', line)-1);
       Delete(line, 1, Pos('#', line));
       rType := StrToInt(line);
       Inc(responseCounter);
       responses[responseCounter] := TResponse.Create(postID, user, content,
                        dt, rType);
      end
     else
      begin
       dt := line;
       Inc(postCounter);
       posts[postCounter] := TPost.Create(postID, user, content, dt);
      end;
    end;

   CloseFile(inFile);
  end
 else
  begin
   ShowMessage('File does not exist');
  end;

end;

// Question 4.4
function TInstaPageManager.getAllPosts: String;
var
 loop : integer;
begin
 Result := '';

 for loop := 1 to postCounter do
  begin
   Result := Result + posts[loop].toString + #13;
  end;
end;
```

Split into tokens

Inserted into correct array
Increment Counter

```
// Question 7.1 continued
 Array of TPostWithResponses objects declared somewhere
function TInstaPageManager.combinePostsAndResponses: String;
var
 tmpResponses : ResponseArray;
 loop, loop2, tempCounter : integer;
begin
 Result := '';
 SetLength(prArray, postCounter);
 for loop := 1 to postCounter do
  begin
   SetLength(tmpResponses, 0);
   SetLength(tmpResponses, 20);
   tempCounter := 0;
   for loop2 := 1 to responseCounter do
    begin
     if posts[loop].getPostID = responses[loop2].getPostID then
      begin
       tmpResponses[tempCounter] := responses[loop2];
       Inc(tempCounter);
      end;
    end;

   SetLength(tmpResponses, tempCounter);
   prArray[loop] := TPostWithResponse.Create(posts[loop], tmpResponses);
   Result := Result + prArray[loop].toString + #13 + #13;
  end;
end;

end.
```


QUESTIONS 5 AND 7.2

```
// Question 5.1
unit uFrmInstaPageUI;

interface

uses
 Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
 Dialogs, StdCtrls, ComCtrls, uInstaPageManager;

type
 TfrmInstaPageUI = class(TForm)
  rchOutput: TRichEdit;
  procedure FormActivate(Sender: TObject);
 private
  { Private declarations }
 public
  { Public declarations }
 end;

var
 frmInstaPageUI: TfrmInstaPageUI;
 // Question 5.2
 ipm : TInstaPageManager;

implementation

{$R *.dfm}

procedure TfrmInstaPageUI.FormActivate(Sender: TObject);
begin
 // Question 5.3
 ipm := TInstaPageManager.Create;
 rchOutput.Lines.Add(ipm.getAllPosts);


 // Question 7.2
 rchOutput.Lines.Add(ipm.combinePostsAndResponses);
end;

end.
```


QUESTION 6

```
unit uPostWithResponse;

interface

uses uPost, uResponse;

type ResponseArray = array of TResponse;

// Question 6.1
type TPostWithResponse = class
 private
  // Question 6.2
  post : TPost;
  responses : ResponseArray;
 public
  constructor Create(inPost : TPost; const inResponses : ResponseArray);
  function toString : String;
end;

implementation

{ TPostWithResponse }

// Question 6.3
constructor TPostWithResponse.Create(inPost: TPost;
 const inResponses: ResponseArray);
begin
 post := inPost;
 responses := inResponses;
end;

// Question 6.4
function TPostWithResponse.toString: String;
var
 loop : integer;
begin
 Result := post.toString + #13 + #9 + #9 + 'Reactions: ';

 for loop := 0 to Length(responses) - 1 do
  begin
   Result := Result + #13 + #9 + #9 + responses[loop].toString;
  end;
end;

end.
```


OUTPUT

SECTION A STRUCTURED QUERY LANGUAGE

QUESTION 1.1

Query1					
ProfileID	**Username**	**Fullname**	**Email**	**Privacy**	**DailyPostRate**
87	cwatson2e	Carolyn Watson	cwatson2e@diigo.com	private	25
12	hadamsb	Henry Adams	hadamsb@vk.com	public	24.7
86	tmorales2d	Teresa Morales	tmorales2d@desdev.cn	public	24.7
59	jgarcia1m	James Garcia	jgarcia1m@cisco.com	private	24.4
1	kmoore0	Kenneth Moore	kmoore0@mashable.com	public	24.2
15	ccoopere	Craig Cooper	ccoopere@yandex.ru	private	23.5
72	cfreeman1z	Charles Freeman	cfreeman1z@seattletimes.com	public	23.1
41	fsimpson14	Fred Simpson	fsimpson14@geocities.com	private	22.4
73	kbowman20	Kevin Bowman	kbowman20@hibu.com	public	22.4
56	jpeterson1j	Jeffrey Peterson	jpeterson1j@adobe.com	public	22.1
91	gwilliams2i	Gregory Williams	gwilliams2i@cyberchimps.com	public	21.7
50	eporter1d	Emily Porter	eporter1d@prnewswire.com	private	21.6
14	mevansd	Matthew Evans	mevansd@skype.com	private	21.3
78	rcox25	Robert Cox	rcox25@smugmug.com	public	21
48	mgarza1b	Margaret Garza	mgarza1b@prweb.com	public	20.9
22	dsullivanl	Denise Sullivan	dsullivanl@rediff.com	public	20.1
31	kstanleyu	Kathryn Stanley	kstanleyu@ox.ac.uk	private	19.9
35	ejordany	Eric Jordan	ejordany@hostgator.com	private	19.4
6	showard5	Sara Howard	showard5@surveymonkey.com	private	19.2
13	emorrisc	Earl Morris	emorrisc@freewebs.com	private	18.9
62	mhart1p	Michelle Hart	mhart1p@berkeley.edu	private	18.9
74	lfernandez21	Larry Fernandez	lfernandez21@hatena.ne.jp	private	18.8
95	ggriffin2m	George Griffin	ggriffin2m@comcast.net	public	18.4
5	jlarson4	Jimmy Larson	jlarson4@cafepress.com	public	18.4
30	jrayt	Juan Ray	jrayt@dedecms.com	private	18.2
29	progerss	Patricia Rogers	progerss@cargocollective.com	private	18.1
32	bmccoyv	Brandon Mccoy	bmccoyv@newsvine.com	public	17.6
36	twoodz	Thomas Wood	twoodz@cafepress.com	public	17.6
28	nhamiltonr	Nancy Hamilton	nhamiltonr@cargocollective.com	private	17.6
99	jburke2q	Jeffrey Burke	jburke2q@loc.gov	public	17.6
83	sdiaz2a	Stephen Diaz	sdiaz2a@seattletimes.com	private	17.5
77	jhudson24	Jessica Hudson	jhudson24@free.fr	public	17.4
98	rellis2p	Raymond Ellis	rellis2p@amazon.co.uk	public	17.3
82	eharris29	Edward Harris	eharris29@simplemachines.org	private	17.3
66	rjames1t	Russell James	rjames1t@addtoany.com	private	17.1
76	jdixon23	Joe Dixon	jdixon23@dailymail.co.uk	private	16.7
51	awalker1e	Adam Walker	awalker1e@xinhuanet.com	private	16.5
71	rlawson1y	Randy Lawson	rlawson1y@livejournal.com	private	16.2
53	jstevens1g	Jesse Stevens	jstevens1g@indiegogo.com	private	16.2
37	gcampbell10	Gary Campbell	gcampbell10@nydailynews.com	private	16.1
44	jmontgomery17	Jeremy Montgomery	jmontgomery17@home.pl	public	15.5
60	sperkins1n	Sarah Perkins	sperkins1n@wix.com	private	15.1
64	rhill1r	Ralph Hill	rhill1r@amazon.de	private	14.9

Query1					
ProfileID	**Username**	**Fullname**	**Email**	**Privacy**	**DailyPostRate**
27	akimq	Ann Kim	akimq@ucsd.edu	public	14.7
21	jriverak	Jose Rivera	jriverak@buzzfeed.com	public	14.4
34	wlawsonx	Willie Lawson	wlawsonx@accuweather.com	private	14.4
7	jtaylor6	James Taylor	jtaylor6@mlb.com	public	14.2
42	ldaniels15	Laura Daniels	ldaniels15@alibaba.com	public	14.1
4	lruiz3	Lillian Ruiz	lruiz3@nhs.uk	private	13.9
65	mmurphy1s	Marilyn Murphy	mmurphy1s@sohu.com	public	13.6
75	grogers22	Gerald Rogers	grogers22@imageshack.us	public	12.2
2	sberry1	Stephanie Berry	sberry1@unesco.org	public	11.9
16	mbellf	Mildred Bell	mbellf@merriam-webster.com	private	11.8
40	tjohnston13	Teresa Johnston	tjohnston13@statcounter.com	private	11.8
97	kmoreno2o	Keith Moreno	kmoreno2o@liveinternet.ru	private	11.2
8	cphillips7	Charles Phillips	cphillips7@google.pl	public	11.1
55	wspencer1i	Walter Spencer	wspencer1i@aol.com	public	11.1
92	pmendoza2j	Paul Mendoza	pmendoza2j@qq.com	public	10.7
17	jreyesg	Joan Reyes	jreyesg@cocolog-nifty.com	private	10.6
10	sortiz9	Shawn Ortiz	sortiz9@csmonitor.com	public	10.1
38	pgordon11	Patricia Gordon	pgordon11@4shared.com	private	9.6
23	charrisonm	Cheryl Harrison	charrisonm@prweb.com	public	9.3
43	rgardner16	Rebecca Gardner	rgardner16@gmpg.org	public	9.3
88	cpayne2f	Craig Payne	cpayne2f@yellowpages.com	private	9.1
46	jmartin19	Jack Martin	jmartin19@zimbio.com	public	9
24	jdixonn	James Dixon	jdixonn@hubpages.com	private	8.9
100	rgreen2r	Roy Green	rgreen2r@engadget.com	private	8.8
96	jflores2n	Jean Flores	jflores2n@lycos.com	private	8.3
47	amason1a	Angela Mason	amason1a@cnet.com	public	7.7
67	hdean1u	Harold Dean	hdean1u@marriott.com	private	7.7
61	pgordon1o	Paul Gordon	pgordon1o@trellian.com	private	7.7
52	jflores1f	Jimmy Flores	jflores1f@loc.gov	public	7
25	njohnstono	Norma Johnston	njohnstono@ycombinator.com	public	7
3	ahernandez2	Ashley Hernandez	ahernandez2@psu.edu	private	6.5
26	mlongp	Mary Long	mlongp@wix.com	public	6
18	tpattersonh	Thomas Patterson	tpattersonh@delicious.com	public	5.9
20	jjohnsonj	Johnny Johnson	jjohnsonj@addtoany.com	private	5.8
80	rfoster27	Russell Foster	rfoster27@mysql.com	private	5.7
33	egrayw	Eugene Gray	egrayw@cbslocal.com	private	5.6
19	grosei	Gregory Rose	grosei@blogspot.com	private	5.5
11	enelsona	Emily Nelson	enelsona@miitbeian.gov.cn	private	4.7
70	dparker1x	Douglas Parker	dparker1x@dedecms.com	private	4.7
68	jwells1v	Janice Wells	jwells1v@europa.eu	private	4.4
54	tdean1h	Todd Dean	tdean1h@usda.gov	private	4.3
45	kjones18	Kathy Jones	kjones18@zdnet.com	public	4.3
58	rbarnes1l	Ryan Barnes	rbarnes1l@usda.gov	private	4.2
89	awelch2g	Adam Welch	awelch2g@skyrock.com	private	4.1
94	mwoods2l	Martha Woods	mwoods2l@about.com	private	4
79	ahughes26	Alice Hughes	ahughes26@google.ru	private	3.9
69	lpalmer1w	Lori Palmer	lpalmer1w@hhs.gov	public	3.8
63	dperez1q	Denise Perez	dperez1q@ow.ly	private	3.4
90	ascott2h	Arthur Scott	ascott2h@rediff.com	public	3.2

Query1					
ProfileID	**Username**	**Fullname**	**Email**	**Privacy**	**DailyPostRate**
93	hford2k	Howard Ford	hford2k@dell.com	public	2.8
39	jnguyen12	Jesse Nguyen	jnguyen12@list-manage.com	public	2.4
57	djordan1k	Diana Jordan	djordan1k@shop-pro.jp	private	2.2
85	jmitchell2c	Jean Mitchell	jmitchell2c@nba.com	private	1.7
9	jfreeman8	Jeremy Freeman	jfreeman8@ovh.net	private	1.6
84	bhenderson2b	Benjamin Henderson	bhenderson2b@stumbleupon.com	public	1.5
49	jlane1c	Jack Lane	jlane1c@pbs.org	private	1.3
81	jford28	Jessica Ford	jford28@pbs.org	public	1.1

QUESTION 1.2

Query2	
Username	**Fullname**
kmoore0	Kenneth Moore
sberry1	Stephanie Berry
jlarson4	Jimmy Larson
jtaylor6	James Taylor
cphillips7	Charles Phillips
sortiz9	Shawn Ortiz
hadamsb	Henry Adams
tpattersonh	Thomas Patterson
jriverak	Jose Rivera
dsullivanl	Denise Sullivan
charrisonm	Cheryl Harrison
njohnstono	Norma Johnston
mlongp	Mary Long
akimq	Ann Kim
bmccoyv	Brandon Mccoy
twoodz	Thomas Wood
jnguyen12	Jesse Nguyen
ldaniels15	Laura Daniels
rgardner16	Rebecca Gardner
jmontgomery17	Jeremy Montgomery
kjones18	Kathy Jones
jmartin19	Jack Martin

Query2	
Username	**Fullname**
amason1a	Angela Mason
mgarza1b	Margaret Garza
jflores1f	Jimmy Flores
wspencer1i	Walter Spencer
jpeterson1j	Jeffrey Peterson
mmurphy1s	Marilyn Murphy
lpalmer1w	Lori Palmer
cfreeman1z	Charles Freeman
kbowman20	Kevin Bowman
grogers22	Gerald Rogers
jhudson24	Jessica Hudson
rcox25	Robert Cox
jford28	Jessica Ford
bhenderson2b	Benjamin Henderson
tmorales2d	Teresa Morales
ascott2h	Arthur Scott
gwilliams2i	Gregory Williams
pmendoza2j	Paul Mendoza
hford2k	Howard Ford
ggriffin2m	George Griffin
rellis2p	Raymond Ellis
jburke2q	Jeffrey Burke

QUESTION 1.3

(No output)

QUESTION 1.4

Query4	
GroupID	**AvgPosts**
1	321.3
2	529.2
3	425.7
4	523.6
5	451.7
6	578.1
7	472.5
8	490.9
9	520.2
10	529.5
11	388.2
12	500.5
13	609.9
14	540.7
15	467.8

QUESTION 1.5

Query5	
CreatedMonth	**TotalGroups**
3	3
10	4

QUESTION 1.6

Query6	
GroupID	**PredictedWeeklyPosts**
1	1138.9
2	1224.3
3	1440.6
4	1565.9
5	1782.2
6	1316.7
7	1293.6
8	560
9	898.1
10	1312.5
11	793.1
12	1624.7
13	1412.6
14	1528.1
15	2170

QUESTION 1.7

Query7	
GroupName	**PrivateAdministrators**
Cool Kids	1
Dank Memes	2
Gossip Groupies	1
Movie Chat	1
Rugby Group	1
Smith Family	2
Tech Gurus	2
The Pi Eaters 314	1
The Untouchables	1
Toys for Boys	1
Ziemann and Sons	1

QUESTION 1.8

(No output)

SECTION B OBJECT-ORIENTED PROGRAMMING

FINAL OUTPUT:

```
2016-12-25 10:45 d0ugSm1th posted: Merry Christmas everyone!
2017-09-25 16:41 s1ckboy77 posted: Loving life!
2017-06-25 00:08 L3xiJones posted: Struggling to sleep at night. Any
advice?
2017-01-25 21:45 GuruZA posted: Trump is such a terrible president!

2016-12-25 10:45 d0ugSm1th posted: Merry Christmas everyone!
           Reactions:
           2016-12-25 10:45 susanSmith89 commented on this post: To you
too Doug!
           2016-12-25 11:02 susanSmith89 loved this post
           2016-12-25 13:21 BobSeegers likes this post

2017-09-25 16:41 s1ckboy77 posted: Loving life!
           Reactions:
           2017-09-25 16:41 LizelleDeBruin commented on this post: You
have plenty to be grateful for.
           2017-09-25 16:44 Kwezi88 commented on this post: Love your
positive attitude.
           2017-09-25 17:49 Kwezi88 loved this post
           2017-09-25 17:49 LeRouxLight likes this post

2017-06-25 00:08 L3xiJones posted: Struggling to sleep at night. Any
advice?
           Reactions:
           2017-06-25 00:08 ReggieSanders53 loved this post
           2017-06-25 08:45 TheSciGuy commented on this post: Try some
rooibos tea before bed.
           2017-06-25 09:54 Ins0mnia commented on this post: If you find
a solution please let me know.
           2017-06-26 03:22 SiyaSandton dislikes this post

2017-01-25 21:45 GuruZA posted: Trump is such a terrible president!
           Reactions:
           2017-01-25 21:45 JumpingJill2 commented on this post: That's
democracy for you.
           2017-01-25 22:45 JumpingJill2 likes this post
           2017-01-25 22:45 ReggieSanders53 commented on this post: Big
business really likes him.
           2017-01-27 21:33 PerfectPete likes this post
           2017-01-27 23:54 Fikile66 dislikes this post
```


ANNEXURE A

ALTERNATE SOLUTIONS FOR QUESTIONS 6 and 7

Alternate Solution #1

QUESTION 6

```
// Question 6.1
public class PostWithResponse
{
  // Question 6.2
  private Post post;
  private Response responses[];
  int counter = 0;

  // Question 6.3
  public PostWithResponse(Post
      inPost, Response[] inResponses, int inCounter);
  {
    post = inPost;
    responses = inResponses;
    counter = inCounter;
  }

  // Question 6.4
  public String toString()
  {
    String rString = post.toString() + "\n\t\tReactions: ";

    for(int i = 0; i < counter; i++)
    {
      rString += "\n\t\t" + responses[i].toString();
    }

    return rString;
  }
}
```

The mark allocated in Question 7.1 stating: “Some algorithm for ensuring that the array of Response objects associated with each Response is of the correct size”. This mark can be awarded for negating the use of a precisely sized array by making CORRECT use of a counter.

QUESTION 7

```
  private PostWithResponse prArray[];

  public String combinePostsAndResponses()
  {
    String rString = "";
    prArray = new PostWithResponse[postCounter];
    for(int i = 0; i < postCounter; i++)
    {
      Response tmpResponses[] = new Response[100];
      int tempCounter = 0;
      for (int j = 0; j < responseCounter; j++)
      {
        if (posts[i].getPostID() == responses[j].getPostID())
        {
          tmpResponses[tempCounter++] = responses[j];
        }
      }

      prArray[i] = new PostWithResponse(posts[i], tmpResponses, tempCounter);
      rString += prArray[i].toString() + "\n\n";
    }

    return rString;
  }

}
```

The mark allocated in Question 7.1 stating: “Some algorithm for ensuring that the array of Response objects associated with each Response is of the correct size”. This mark is awarded for the use of a counter in the **PostWithReponse** class.

Alternate Solution #2

QUESTION 6

```
// Question 6.1
public class PostWithResponse
{
  // Question 6.2
  private Post post;
  private Response responses[];


  // Question 6.3
  public PostWithResponse(Post inPost, Response[] inResponses);
  {
    post = inPost;
    responses = inResponses;
  }

  // Question 6.4
  public String toString()
  {
    String rString = post.toString() + "\n\t\tReactions: ";

    int i = 0;
    while (responses[i] != null && i < responses.length)
    {
      rString += "\n\t\t" + responses[i].toString();
      i++;
    }
    return rString;
  }
}
```

The mark allocated in Question 7.1 stating: “Some algorithm for ensuring that the array of Response objects associated with each Response is of the correct size”. This mark can be awarded for checking for null values and stopping the loop once null values are found in the responses array.

QUESTION 7

```
  private PostWithResponse prArray[];

  public String combinePostsAndResponses()
  {
    String rString = "";
    prArray = new PostWithResponse[postCounter];
    for(int i = 0; i < postCounter; i++)
    {
      Response tmpResponses[] = new Response[100];
      int tempCounter = 0;
      for (int j = 0; j < responseCounter; j++)
      {
        if (posts[i].getPostID() == responses[j].getPostID())
        {
          tmpResponses[tempCounter++] = responses[j];
        }
      }

      prArray[i] = new PostWithResponse(posts[i], tmpResponses);
      rString += prArray[i].toString() + "\n\n";
    }

    return rString;
  }

}
```

The mark allocated in Question 7.1 stating: "Some algorithm for ensuring that the array of Response objects associated with each Response is of the correct size". This mark can be awarded for checking for null values and stopping the loop once null values are found in the responses array property in the **PostWithResponses** class.

Alternate Solution #3

NOTE: This solution makes use of inheritance. Therefore there may be changes to the original **Post** class. This could include

- Making attributes protected
- Adding accessor methods for all properties and fields

DO NOT DEDUCT MARKS FOR QUESTION 3 IF ANY OF THESE CHANGES OR ADDITIONS HAVE BEEN MADE IN THE Post CLASS

QUESTION 6

```
public class PostWithResponse extends Post
{
  // Question 6.2
  private Response responses[];

  // Question 6.3 – Constructor with Post object
  // 4 marks
  public PostWithResponse(Post inPost, Response[] inResponses)
  {
    super(inPost.getPostID(), inPost.getUser(), inPost.getPostContent(),
              inPost.getPostDateTime());
    responses = inResponses;
  }

  // Question 6.3 Alternative – Constructor with basic data types
  public PostWithResponse(int inID, String inPostUser, String
                          inPostContent, String inPostDateTime,
                          Response[] inResponses)
  {
    super(inID, inPostUser, inPostContent, inPostDateTime);
    responses = inResponses;
  }

  // Question 6.4
  public String toString()
  {
    String rString = super.toString() + "\n\t\tReactions: ";

    for(int i = 0; i < responses.length; i++)
    {
      rString += "\n\t\t" + responses[i].toString();
    }

    return rString;
  }
}
```


QUESTION 7

```
  // Question 7.1
  private PostWithResponse prArray[];

  public String combinePostsAndResponses()
  {
    String rString = "";
    prArray = new PostWithResponse[postCounter];
    for(int i = 0; i < postCounter; i++)
    {
      Response tmpResponses[] = new Response[100];
      int tempCounter = 0;
      for (int j = 0; j < responseCounter; j++)
      {
        if (posts[i].getPostID() == responses[j].getPostID())
        {
          tmpResponses[tempCounter++] = responses[j];
        }
      }

      Response postResponses[] = new Response[tempCounter];
      System.arraycopy(tmpResponses, 0, postResponses, 0, tempCounter);
      prArray[i] = new PostWithResponse(posts[i], postResponses);

      // If alternate constructor was used
      //prArray[i] = new PostWithResponse(posts[i].getPostID(),
                        posts[i].getUser(), posts[i].getPostContent(),
           posts[i].getPostDateTime(), postResponses);

      rString += prArray[i].toString() + "\n\n";
    }

    return rString;
  }

}
```